METHOD AND SYSTEM FOR HIGH-SPEED SOFTWARE RECONFIGURABLE CODE DIVISION MULTIPLE ACCESS COMMUNICATION

Related Application

This application is a continuation application of International Application PCT/BE00/00053 filed on May 10, 2000 and published in English on November 16, 2000, which claims priority to U.S. provisional application number 60/133,340 filed on May 10, 1999.

Background of the Invention

Field of the Invention

The present invention is relates to a communication device for W-CDMA and a method of operating the communications device.

Description of the Related Technology

A communication device, for example, for Wideband Code Division Multiple Access (W-CDMA) is configured to operate in accordance with a predetermined telecommunication standard and predetermined parameters. The communication device typically has a digital signal processor (DSP) that allows it to adapt to changes of the physical layer.

There is therefore a need for a W-CDMA system, which allows the implementation of various telecommunication standards, and various applications realizable according to these standards, without the need for a powerful DSP processor for the flexible part of the physical layer. Further, there is a need for a W-CDMA apparatus that provides for various fading channel circumstances.

Summary of Certain Inventive Aspects

One aspect of an inventive embodiment involves a communication device for W-CDMA signal transmission and reception. The communication device has a W-CDMA transmitter having at least one of a first RAM and first registers, wherein the transmitter is configured to operate in accordance with first parameters. Further, the communication

device has a W-CDMA receiver having at least one of a second RAM and second registers, wherein the receiver is configured to operate in accordance with second parameters, and a signal acquisition component. A processor is in communication with the W-CDMA transmitter, the W-CDMA receiver and the signal acquisition component, and configured to provide for software configuration of the first and second parameters.

A software reconfigurable component for which parameters of a circuit and/or algorithmic alternatives for this circuit may be configured using software settings. The circuit itself is built up of logic, and contains memory, such as registers, a RAM, or both, which are preferably controlled by a processor subsystem, which performs the above mentioned software settings. Such an approach leads to lesser power consumption if compared to a complete software implementation, while there is still sufficient flexibility possible.

The communication device may further comprise a data processor. Such a processor may be any kind of processor capable of changing the settings of the device. Examples of such processors are DSP processors, microprocessors, microcontrollers, FPGA, logic circuits and FSM circuits.

In one embodiment of the communication device, the processor is configured to reconfigure the communication device. The processor may control the RAM, the resisters, or both, of the W-CDMA signal transmitter and receiver. The transmitter may have a first programmable pulse shaping filter and the receiver may have a second programmable pulse shaping filter which may be programmable to perform GMSK filtering while the transmitter and receiver are configured to interface with a GSM front-end. The processor may be configured to perform the GSM protocol stack.

In one embodiment, the communication device is configured for waveform transmission, reception, acquisition, or a combination thereof, of signals selected from the group consisting of UMTS, Satellite UMTS, Galileo, GPS, IS-2000, IMT-2000, CDMA2000, IS-95, 3GPP, 3GPP2 and ARIB signals.

In one embodiment, the transmitter of the communication device may include at least one of a synchronization hardware to slave transmit start epochs to events external to the transmitter, a burst generator for realizing discontinuous transmissions, a QPN channel

containing one or more spreaders with their own amplification of the output, a combiner to accumulate the QPN channel output, a PN code generator, a scrambling code generator, a scrambler, a combiner which accumulates the scrambling code output, a pulse shaping oversampling filter, and an NCO and upconverter for carrier precompensation.

The PN code generator may be realized as a RAM in which the PN codes are downloaded under control of the processor. The scrambling code generator may be realized as a programmable Gold Code generator. The QPN channel may be configured to execute UMTS forward or return link transmission. The amplification of the spreader output may be configured to perform transmit power control.

In one embodiment, the transmitter of the communication device has a time interpolator to perform sub-chip time alignments (for example, for S-CDMA). Further, the transmitter of the communication device may be configured for multi-code transmission.

In one embodiment, the receiver of the communication device may include a pulse shaping filter, an optional level control block, a demodulator assigned to track the multi-path components received from one base station, and a reference demodulator for S/(N+I) measurements. Further, the receiver may have a downconverter prior to the pulse-shaping filter in order to interface at a front-end at an intermediate frequency. The receiver may also be configured for execution of at least one of the following signal protocols: UMTS, Satellite UMTS, Galileo, GPS, IS-2000, IMT-2000, CDMA2000, IS-95, 3GPP, 3GPP2, and ARIB forward link and return link waveforms.

In one embodiment, the level control block may have a programmable shifter to perform coarse grain dynamic control, a programmable multiplier to perform fine grain dynamic control, an overflow counter operating on the most significant bit and the second most significant bit, an overflow counter operating on the second most significant bit and the third most significant bit, and a saturation logic to clip the result from the multiplier. The level control block may be operated in a runtime control loop by the processor.

In one embodiment, the demodulator may have a Rake filter producing a signal at a chip rate which is a coherent accumulation of channel corrected multi-path components resulting from one base station, and a tracking unit using the signal at chip rate for descrambling and despreading a plurality of waveform channels.

The Rake filter has a FIFO to buffer samples at a chip rate coming from the level control block, a delay line containing a plurality of registers, wherein the input of the delay line is connected to the output of the FIFO, a plurality of finger blocks having inputs connected to programmable tap positions on the delay line, and a summator of complex outputs of the finger blocks at chip rate. The finger blocks are preferably respectively grouped in a "late" multi-path group and an "early" multi-path group. The Rake filter is configured to accumulate the energies of the outputs of the late multi-path group and the early multi-path group, and to use these accumulated values to feed the time error detector of a DLL used for time tracking.

In one embodiment, the Rake filter may include memories to hold one or more of spreading code for a channel correction pilot, scrambling code for a channel correction pilot, a channel correction pilot symbol modulation, a channel correction pilot symbol activities. The memories may be controlled by the processor.

In one embodiment, the finger block may have a channel correction pilot descrambler, a channel correction pilot despreader, and a channel correction pilot filter, which first performs a coherent channel correction pilot symbol accumulation over a programmable number of steps, and which secondly produces a weighted average on a programmable number of the coherent channel correction pilot symbol accumulation over a programmable number of steps. Further, the finger block has a channel estimator generating a channel estimation at a chip rate using the outputs of the pilot filter, a channel corrector performing a multiplication of the incoming chip stream with the complex conjugate of the channel estimation, a calculation of the slot energy, a comparison of the slot energy with a programmable threshold, a circuit to force the channel estimation to zero if the threshold is not exceeded.

The finger may be configured for slow and fast fading compensation, for example, by programming the channel correction pilot filter for slow fading. The channel correction pilot filter first performs a coherent accumulation over a slot, and secondly performs a weighted average over previous-previous, previous, actual and next obtained slot values. This yields a channel estimation per slot, which is applied by the channel corrector. For fast fading, the channel correction pilot filter first performs a coherent accumulation over a slot, and then

derives channel estimations through interpolating consecutively the coherent accumulations over a slot. This yields in channel estimations with sub-symbol timing which are applied by the channel corrector.

In one embodiment, the reference demodulator may have an accumulator of programmable length of the absolute values of samples at a chip rate, and a low pass filter operating on the accumulator output. The reference demodulator may be configured to operate in a runtime control loop by the processor. Further, the demodulator may be configured to perform satellite diversity.

In one embodiment, the communication device may be configured to perform accurate ranging measurements to geostationary satellites. The communication device may be implemented in an integrated circuit. Further, the communication device may be implemented within an intellectual property core (as a building block for inclusion in an integrated circuit).

Another aspect of a one embodiment involves a method of operating a W-CDMA communication device. The method configures the communication device for a predetermined use, and transmits, receives, acquires, or a combination thereof, waveform signals. The waveform signals may be selected from the following: UMTS, Satellite UMTS, Galileo, GPS, IS-2000, IMT-2000, CDMA2000, IS-95, 3GPP, 3GPP2 and ARIB signals. The configuring may be performed by a processor.

Brief Description of the Drawings

These and other aspects, advantages, and novel features of the invention will become apparent upon reading the following detailed description and upon reference to the accompanying drawings. In the drawings, the same elements have the same reference numerals.

Figure 1 represents a global transmitter structure of one embodiment of a communication device.

Figure 2 represents a QPN channel.

Figure 3 represents the use of a RAM block to generate PN-codes.

Figures 4 to 7 represent possible RAM 5 configurations for the communication device.

Figure 8 represents one embodiment of a receiver architecture for the communication device.

Figures 9 represents one embodiment of a level control for the communication device.

Figure 10 represents one embodiment of a noise estimator for the communication device.

Figure 11 represents one embodiment of a general overview of a demodulator for the communication device.

Figure 12 represents one embodiment of a tracking unit for the communication device.

Figure 13 represents one embodiment of a demodulator for UMTS mode, using only one tracking unit.

Figure 14 represents one embodiment of a Rake receiver.

Figure 15 represents one embodiment of a Rake finger.

Figure 16 represents one embodiment of a slotwise coherent pilot symbol accumulation.

Figure 17 represents one embodiment of a finger energy calculation.

Figure 18 represents one embodiment of a slot weighing filter for the communication device.

Figure 19 shows an exemplary overview of a Rake finger process for Channel Mode 0.

Figures 20 and 21 show an exemplary overview of a Rake finger process for Channel Mode 1.

Detailed Description of Certain Inventive Embodiments

In the following detailed description, the following abbreviations are used:

BS	Base station
CCPCH	Common Control Physical Channel
DL	Downlink
DPCH	Dedicated Physical Channel
DPCCH	Dedicated Physical Control Channel
DPDCH	Dedicated Physical Data Channel

GSM	Global System for Mobile communication
HO	HandOver
MRC	Maximum Ratio Combining
MS	Mobile station
OVSF	Orthogonal Variable Spreading Factor
PN	Pseudo-Noise
PRACH	Physical Random Access Channel
QPN	Quadrature Pseudo-Noise
RSSI	Received Signal Strength Indication
SF	Spreading factor
UL	Uplink
W-CDMA	Wide Band CDMA

Transmitter Specification

The global structure of an exemplary transmitter 1 is shown in Figure 1 and explained in detail hereinafter.

QPN channels with synchronization hardware and PN-code generators

The transmitter 1 contains a plurality of QPN channels 3 as shown in Figure 2. These channels are, for example, combined in two sets of four QPN channels (set A and set B) and a set C with only one QPN channel, as shown in Figure 1. Each set has a separate block for generating a PN-code 5 and a separate synchronization hardware 7, which defines a start of symbol transmission.

Synchronization hardware

An output of the synchronization hardware goes to the QPN channels of a set and defines a common symbol start moment for all QPN channels in a set. This signal is generated as a selection of one out of a plurality of incoming signals with a programmable offset. The incoming sync channels may, for example, be generated by: another chip, TX timers, receiver pulse, acquisition hardware output, or the like. In one embodiment, a counter

at the chip rate may be used to generate the offset. This gives an offset resolution of one 'primary' chip. The range of the offset is [0:65535]. This is sufficient to give an offset of one frame for UMTS (40960 chips).

QPN channel 3

Each QPN channel 3 has the functional structure represented in Figure 2. Its functional elements are described below:

Spreader 11

Input binary symbols coming directly from an interface (symbI 13 and symbQ 14) are spread with PNbits PNbitI and PNbitQ. Each symbol has an activity bit (actI and actQ). When this is 0 the functional spreader output will be 0 instead of +1 or -1. This activity bit is used for burst transmission and for BPSK instead of QPSK/QPN transmission. Signals symbI and actI are signals at a symbol rate fsIxx. Signals symbQ and actQ are signals at a symbol rate fsQxx. The symbol rate fsIxx may differ from the symbol rate fsQxx. The spreading factor is set by a sfI input 15 and a sfQ input 16. The spreaders may be (re)started via a sync signal 17. A rate fcp is defined as: fcp=fsIxx * sfI = fsQxx * sfQ.

Symbol clock signals 19 (symbclkI and symbclkQ) are generated as a symbol reference for other hardware that requires symbol synchronous actions, like the gain controls 21.

Gain Control (Transmit power control)

Each complex spreader 11 is followed by a separate gain control 21. Each output branch of a spreader is again separately gain controlled.

PN-code generators

The PN-code generators generate complex PN codes for the QPN channels 3 (Figure 1). A code generator 5 is provided for a set. For example: the PN-code generators 5 for sets A and B generate each four complex codes, while the PN-code generator 6 for set C generates only one complex PN-code.

Gold code generator

This is a classical Gold code generator with, for example, 42 bit registers which may generate any Gold code with any length up to (2^{42})-1. It may also be used to generate any segment out of a Gold code smaller than (2^{42})-1.

The sgfb inputs define the feedback position in the shift register and the init inputs are used to initialize the shift registers at reset or restart. The poly inputs are used to program the polynomials to generate the Gold sequences. The rest signals are used to generate a small section of the complete Gold code and then jump to the back to the init value. If the register in the gold code generator reaches the rest state, the register is in the following clock-cycle re-initialized.

RAM based code generation

Each set has a block 23 that may generate PN-codes based on a RAM. For all three sets the same block 23 is used. This is shown in Figure 3. The block 23 contains a RAM of, for example, 8*1024 bits. An address generator 31 selects one row 35 of this RAM with the x address. These 8 bit are then routed to the spreaders via a switch controlled by an address y. The address generator 31 has a start 25, stop 27 and step input 29. The address generator 31 may be configured in different ways with the configure input 33. It is possible to stop the generators when the activity bit of a symbol is 0.

The following are examples of possible RAM configurations:

- Figure 4: 8 BPSK streams 37, streams 0, 1, 4, 5, 6 and 7 have SF 1024, stream 2 has SF 512 and stream 3 has SF 256; x counts from 1023 to 0, y is a static value.
- Figure 5: 6 BPSK streams 37, stream 0 and 5 have SF 2048, stream 1 has SF 512, stream 2 has SF 256, streams 3 and 4 have SF 1024; x counts from 1023 to 0, y changes between two values every 1024 chips.
- Figure 6: 2 BPSK streams 37, stream 0 and 1 have SF 256, stream 0 uses continuously the same code while stream 1 uses a sequence of 16 different codes. This scheme is usable for SCH transmission if the address counter is stopped

when the activity bit is 0. x counts from 1023 to 0, y changes between 4 values every 1024 chips.

- Figure 7: 4 BPSK streams 37, stream 0 has SF 1000, stream 1 has SF 2000, stream 2 has SF 400, stream 3 has SF 600; x counts from 999 to 0, y changes between 3 values every 1024 chips.

As shown in these examples, in the case of variable spreading factor transmission (for example, OVSF codes in UMTS), it is assumed that spreading factors have a common multiple. The RAM is filled with replicas until the common multiple length is reached. In this way the symbols in one set are multiple-symbol synchronous.

Combiners at fcp rate

The two combiners 38 after set A and set B at fcp rate output the sum of the 4 incoming complex numbers.

Scrambler 40 and scrambling code generation

Scrambling code generator 41

The scrambling code generator 41 block generates the complex scrambling code Cscramb = cI+jcQ. Each scrambling code generator has its own synchronization hardware block to generate the sync signal. (see Figure 1). The scrambling code generator 41 contains two Gold code generators with 42 bit register, two RAMs of 256 bit, an interface for external input of codes and extra hardware for UMTS to modify the Gold codes. The Gold code generators are functionally the same as the Gold code generators in the PN code generators. A classical Gold code generator with 42 bit registers may generate any Gold code with any length up to (2^{42})-1. It may also be used to generate any segment out or a Gold code smaller than (2^{42})-1.

The sgfb inputs define the feedback position in the shift register, the init inputs are used to initialize the shift registers at reset or restart. The poly inputs are used to program the polynomials to generate the Gold sequences. The rest signals are used to make generate a small section of the complete Gold code and then jump to the back to the init value. If the

register in the Gold code generator reaches the rest state, the register is in the following clock-cycle re-initialized. It is possible to re-initialize the generators after a programmable number of chips or to let them run freely.

Examples of different modes

Mode 0

cI and cQ are any Gold code with any length of maximum (2^{42})-1.

Mode 1

cI and cQ come directly from a RAM of 256 bit. It must be possible to use only the first k bits in the RAM, with k smaller than 257.

Mode 2

Mode 0 but with zero extension in front of the generated Gold codes. cI = <0, c1>, cQ = <0, c2>

Mode 3 (UMTS specific)

Mode 0 or Mode 1 but c1 and c2 coming from the Gold code generators or RAM are modified in the following way:

Cscramb = cI+jcQ = c(w + jc'w)

where w0 and wl are chip rate sequences defined as repetitions of:

w = {1 1},

w = {1 –1},

and where c is a real chip rate code, and c' is a decimated version of the real chip rate code. The preferred decimation factor is 2, however other decimation factors should be possible in future evolutions of UMTS if proven desirable.

With a decimation factor of decim=2, c' is given as:

c'(2k)=c'(2k+l)=c(2k), k=0, 1, 2, ...

c1 and c2 are constructed as the position wise modulo 2 sum of 40960 chip segment of two binary m-sequences generated by means of two generator polynomials of degree 41.

The code c2, used in generating the quadrature component of the complex spreading code is a 1024-chip shifted version of the code c1 used in generating the in-phase component.

Scrambler 40

The scrambling is in fact an overlay spreading without changing the chip rate. The change in chip rate is done with the Hold 1-256 block.

Input data : dI+jdQ

Input scrambling code : cI+jcQ

This scrambler 40 has 3 modes:

- Off: output = input
- Complex scrambling: output = (dI+jdQ)*(cI+jcQ) = dI*cI-dQ*cQ + j(dI*cQ+dQ*cI)
- Dual real scrambling: output = dI*cI + j dQ*cQ

Interpolator with chip phase control

The interpolator is used to do a chip phase shift with a resolution smaller than one chip. For every sample input, one output sample is generated, wherein the input and output clock is the equidistant clock. A linear interpolation is used to perform a function:

out(k) = (1-TXMU)*in(k-1)+TXMU*in(k)

where in(k-1) and in(k) are two consecutive equidistant complex samples at rate; and where TXMU is an input of the interpolator and is a number (0<= TX_MU<=1).

Upsampling and programmable filter

The fixed upsampling with a factor of, for example, four (zero insertion) and a symmetrical programmable filter are realized as a complex oversampling polyphase filter. The output sampling rate f4c is: f4c = 4 * fc.

Offset modulation

By setting offset to 1, the Q branch will be delayed with 0.5 chip.

Complex upconvertcr 42 and NCO 44

NCO 44

The NCO 44 generates a cosine and sine value. The cos and sin values are frequency and phase controllable. The specifications below are not required for cellular, but may be used for satellite applications with demanding phase noise requirements. The sine and cosine values are generated with the 16 MSB of a s<32, 0> phase value. The 14 LSB of this 16 bit number go to two lookup tables which contain the values for sin and cos in [C, 2*pi[, with a gain of 2047/2048. The lookup word length for sin and cos in quadrant 1 is u<11, 11>. The 2 MSB of the s<32, 0> bit phase register are used to recover the quadrant, wherein sin and cos are s<12, 11> numbers. The output of the NCO 44 is the complex signal (cos + j .sin).

The s<32, 0> bit phase register may be directly controlled via the TXPHASE input (s<32, 0>) or by integrating with wrap around the TXINC (s<32, 0>) value. The TXINC may be used to program the frequency of the generated sine and cosine in the following way: fsin = fcos = $TX\_INC/2^{32} * f4c$.

With TXINC negative a negative (complex) IF will be generated. For example, to generate a complex carrier at -20 MHz, TXINC should be set to -1073741824. The s<32, 0> phase register should be a part of the chip boot chain.

Upconverter 42

Here a complex upconversion with the NCO 44 generated complex carrier is done. The computations are done full precision, wherein the ten multiplications have one redundant bit as the most negative number will never be present in the sin or cos value. Thus the result of the multiplications are s<32, 24> bit numbers. This makes the full precision outputs bit numbers. These full precision numbers are reduced to s<35, 16> numbers.

Level Control 2

The purpose of the level control 2 is to condition the signal coming from the upconverter prior to the DA conversion.

Receiver Specification

The global receiver structure is shown in Figure 8. All functional blocks are discussed in more detail in the next paragraphs.

Common downconverter with NCO 47

Downconverter 45

The downconverter 45 performs a complex downconversion, with the NCO generated complex carrier, on the incoming complex signal. The output signal is expected to be a near baseband signal.

DO_MODE	data in	carrier in	output
00	X+jY	cos+jsin	(X+jY) * (cos+jsin)
01	X+jY	cos+jsin	(X+jY) * (cos-jsin)
10	X+jY	cos+jsin	X* (cos+jsin)
11	X+jY	cos+jsin	X* (cos-jsin)

Input and output are at fin rate.

Programmable FIR filter 49 with downsampling 51

The complex receive stream coming from the downconverter is filtered by a programmable symmetrical FIR filter and downsampled with a factor RXD. RXD may be 1 or 2. Inputs are at fin rate, outputs at f2ct rate.

Level control 53 with overflow detectors

To optimize the number of significant bits going into the demodulator correlators a common level control is provided to adapt the level of the signal coming from the filter (see Figure 9 for the structure).

The incoming complex data is shifted over RXSHIFT bits 55. This is a coarse gain with 6 dB steps. A lower resolution gain control is done by the multiplication by RXMULT 57. The multiplication is followed by a saturation logic (on the data) and overflow counters. For this reason, the result from the multiplication is extended with 1 MSB to produce the

input for overflow counter 1. Overflow counter 1 59 counts the real overflows, so the overflows where the saturation logic saturates the signal. Overflow counter 2 61 is required to count the overflows as if the signal amplitude was twice as big.

S/(N+I) estimator 63

The noise estimator 63 (Figure 10) provides a filtered complex noise correlation value which may be read by the microcontroller subsystem. This value could be used for setting thresholds in the acquisition hardware. The noise correlator 65 is just the accumulation of NC_length absolute values 64 of the complex input. In this way, an RSSI estimation is obtained. The filter is a hardware low-pass filter. By setting the bypass to 1, the low-pass filter may be bypassed.

Demodulator 67

In most modes the plurality of demodulators are used to support base station diversity for soft handover, however they may also be used for other purposes. In the following paragraphs the demodulator structure is explained in more detail.

Figure 11 is a general overview of a demodulator 67. The demodulator 67 has a number of tracking units 69 with peripheral hardware such as code generators and feedback signal generators such as PED with PLL 70, TED with DLL 97, AED 91 with AGC 93, as discussed below. Each demodulator 67 has a Rake block 71 performing a combination of channel corrected multi-path components. This block is discussed below in more detail.

It is contemplated that not all the hardware in Figure 11 is used at the same time. This depends on the configuration. It is possible to turn off idle blocks to save power.

Tracking unit 69

Each of the, for example, three tracking units 69 (Figure 12) has the same input: the complex signal coming from the common level control. It is possible to track one signal source with one tracking unit. A signal source may be a physical transmitter or it may be a multi-path component coming from one transmitter. So in one demodulator we may, for example, track three satellites or track three multi-path components (as an alternative to the

use of the above mentioned Rake) from a terrestrial base-station. The functional blocks within a tracking unit are described below.

Tracking unit down converter 45 and NCO 47

This block is used as an actuator for the carrier phase/frequency tracking. A final downconversion is performed.

Tracking unit interpolator 74 with chip frequency control

The tracking unit interpolator 74 is used as actuator for the chip phase/frequency tracking. This is done by a pseudo-chip rate chance. The outputted chip rate is controlled via the DINT input. Linear interpolation between samples spaced approximately 0.5 chip is performed by:

out (k) = (1-INTMU)*in(k-1)+INTMU*in(k),

where in(k-1) and in(k) are two consecutive equidistant samples at f2c rate.

The DINT input is used to change the INTMU continuously by adding DINT to the previous value of INTMU every cycle. This results in a change in chip rate by 1/(1+DINT).

- INTMU in [0:1[: one input sample produces one output sample,
- when INTMU 0: two output samples are produced for one input sample, and INTMU is wrapped back into [0:1[,
- when INTMU >= 1: no output sample is produced for one input sample, and INTMU is wrapped back into [0:1[.

The tracking unit interpolator 74 causes a delay of one sample. For example, when DINT = cte = 0, out = in z^{-1} with a 0.0 added at the start.

The input samples are equidistant at f2c rate. The output samples of the interpolator 74 are not equidistant at f2cr rate, which is between f2c/2 and 2*f2c. So all the hardware after the interpolator 74 must be designed to work at 2*f2c although the nominal rate is f2c.

MEL gate 7 5

The MEL gate 75 is used in no-cellular modes; otherwise the MEL gate 75 is bypassed through the appropriate multiplexer settings. The incoming stream at f2cr is split in three streams at f2cr rate.

$E = in.z^{-2}$

$M = in.\ z^{-1}$

$L = in$

In this way each stream is spaced 0.5 chip. The M signal of Tracking unit 0 is also used as input for the Rake block, if it is activated.

Downsampling 80 factor

A phase controllable downsampling with a factor, for example, 2 is performed by skipping one incoming sample of two incoming samples. D2 defines which phase to skip. The output rate is fc = f2cr/2.

Chip stream selection

The three multiplexers 81 allow to chose between which signal goes to the final correlators 83. This can be the downsampled signal coming from the MEL gate 75 or it can be the Rake output at the chip rate.

Scrambling code generator 41

This is functionally the same as the transmitter scrambling code generator, but at a fc rate.

Descrambler 83

Input data: dI+jdQ

Input scrambling code: cI+jcQ

This block should have three modes:

- Off: output = input,
- Complex descrambling: output = (dI+jdQ)/(cI+jcQ) = (dI.cI+dQ.cQ + j(-dI.cQ+dQ.cI))/2.0,

- Dual real scrambling: output = dI*cI + jdQ*cQ.

In the three modes, the delay between input and output should be the same. Input and output are at fc rate.

Despreaders 85

Each tracking unit contains a number of QPN despreaders 85. Each despreader 85 and each branch of the despreader 85 can have a different spreading factor.

Variable amplifiers 87

A variable amplifier 87 is used as an actuator for the signal amplitude tracking. Each variable amplifier 87 (Vamp) can have a different gain. The output of the Vamps 87 are the soft symbols MD, MP, EP and LP which stands for Middle Data, Middle pilot, Early pilot and Late pilot. But when in Rake (UMTS mode), these signals have completely different meanings than these names suggest.

PN-code generators 89

The PN-code generators 89 generate the complex PN-codes for the despreaders 85. This is a similar block as in the transmitter. It is possible to use a RAM, a Gold code generator or an external input.

The tracking unit 0 is equipped with, for example, four separate generators, wherein unit 1 and 2 have only one generator. The four despreaders in unit 1 and 2 use the same despreading code.

AED 91 and AGC 93

The AED 91 is the error detector for the signal amplitude tracking. The AGC 93 provides for a filtering on this signal and outputs the signal going to the variable amplifiers 87. The tracking unit 0 has a separate AED and AGC or each despreader in the tracking unit, while tracking unit 1 and 2 only have a common AED and AGC working on the MP signal.

PLL 70

The NCO of each tracking unit can be set by an external block like ARM software or can be controlled by the PLL. The PLL works on the MP signal. When the Rake is used, the PLL is turned off.

TED0, TED1 and DLL 97

The TED0 or TED1 are used as error detectors for the chip timing tracking. TED1 is used when the CCP is used as a signal source for the despreaders of the unit, while TED0 is used when a classic Early-Late correlator tracking is done. The output of the TED 91 goes to the DLLs and controlling the chip frequency of the interpolator.

Symbol combiner (not shown)

When the three tracking units are used for tracking different multi-paths of the same signal, a hardware combination of the three CD outputs can be performed. Functionally, this is only an addition of the complex CD numbers. However, the symbol timing of CD[0], CD[1] and CD[2] will be different which will complicate tie coherent symbol combining.

Rake receiver 101

The Rake receiver 101 (Figure 14) performs a weighted coherent combination of a plurality of taps selected on a delay line of the chip stream resulting in one new chip stream. To combine the taps weighted coherently, a channel estimation (amplitude, phase) of each of the delayed chip streams is made, as explained below in more detail.

Demodulator using Rake

This section gives a detailed explanation on the use of the demodulator as a receiver where multi-path components are coherently combined at chip rate. The Rake block of the demodulator is only used in this mode and is also discussed in detail in this section. This specification is, for example, for reception of a UMTS waveform. A possible configuration of a demodulator for UMTS mode using only one tracking unit 90, is shown in Figure 13.

The Rake-based demodulator configuration reuses almost everything from the tracking unit except for the PLL and PED. A large extra block that is not used when using Early-Late correlator tracking is the Rake 71. That is, the Rake-based demodulator includes the Rake that generates a new chip stream from the incoming chip stream and the classic descrambler 83, despreader 85 hardware.

With the configuration shown in Figure 13 it is possible to receive four QPN channels. These channels must be synchronous as they use the same Rake receiver. These four QPN channels must also have the same scrambling code. With tracking unit 1 and 2 and with Rake as input, two extra QPN channels with a different scrambling code are received. The extra QPN channels must still be synchronous with the other channels. To receive two asynchronous transmitters, the two demodulators must be used.

The only despreading in the Rake is the pilot symbol despreading used to make the channel estimations. Chip phase tracking is done by a timing error detector (TED0) and DLL working at slot rate.

Rake overview

This part (Figure 14) performs the coherent combination of a plurality chip streams 107 into one new chip stream 109. For example, eight fingers 111 are used, where a channel estimation is done for that chip phase with the aid of pilot symbols. This channel estimation is used to 'correct' the chip stream of the respective finger, after which all fingers can be combined. MRC with optional zero forcing is used to combine the different chip phases. The pilot symbols can have a SF from 4 to 256 and may be arbitrarily distributed over the slot.

Fingers 0 to 4 contribute to the Late multi-paths, fingers 5 to 7 to the Early multi-paths. Note that there is no real 'Middle' finger. This means that in the case of a single path, the correlation energy will be split over fingers 4 and 5 and one will never correlate at the 'top' of the correlation shape.

In one embodiment, the Rake is initialized so that the strongest peak is between fingers 4 and 5. With the phase controllable decimation (D2) the chip phase can be set with a resolution of 1/2 chip. Each finger has as inputs:

- Pcb: codebit for despreading the pilot chip stream. The spreading code is stored in a RAM of 256 bits. This is a real signal, no QPN pilot is possible.
- Psb: complex descrambling bits coming from the descrambling code generator.
- Psy: data modulation on the pilot symbols. One can use a RAM to store the modulation of a complete slot, so one needs a RAM of 640x2 bits. When a higher SF is used not all 640 locations will be used. For example, with SF 256 only the first ten locations of the RAM will be used. pilot modulation can change on a slot-by-slot basis.
- Pac: activity bit for pilot symbols. This eliminates the need for having the pilot portion as a continuous portion at the beginning of the slot. Again a RAM of 640x1 could be used.
- Psf: The pilot SF.
- Chm: channel mode parameter selects the algorithm to use to make the channel estimations. (slow fading: 0, fast fading: 1).

Other configuration inputs include: a threshold to decide on which finger there is a signal, filter coefficients for channel estimation filtering, etc. The RAMs of 640 bits could be smaller if it is not required to have a burst of pilot chips equal to four chips anywhere in the slot. For example, eight consecutive pilot symbols SF can be replaced by one pilot symbol with SF 32.

Each finger has a complex CCCP[x] output at chip rate. This is the delayed chip multiplied with the complex conjugate of the channel estimation of finger x. Each finger also has a FNx output at slot rate which is the energy of the coherent accumulation of all pilot chips/symbols in a slot of finger x. The sum of all FNx is calculated and goes to the pilot AGC. In this way, CCCP is not dependent on the pilot energy.

As there is fixed finger spacing, only a global DLL is needed. The DLL works on the slot rate. The Late and Early energies are calculated as:

ENL = FN0+FN1+FN2+FN3+FN4,

ENE = FN5+FN6+FN7.

ENL and ENE go to the DLL which feeds backs to the interpolator at the input of the demodulator using the Rake filter.

Rake finger 115

This section describes the fingers architecture (see Figure 15).

Descrambler 117

The incoming chips are descrambled with Psb. This code and its phase are common for all fingers. The phase has to be set during an acquisition process initializing the Rake. The descrambler 117 has the same functionality as the other descramblers.

Complex pilot despreader 119

The complex signal coming from the descrambler 117 at the chip rate is despread with the pilot Pncode (Pcb), only one despreader, so the pilot must be a QPSK or BPSK signal. The pilot PNcode has a PNlength of Psf, wherein 4<=Psf<-256, and k*Psf=2560 with k being a positive integer. The despreader 119 works continuously and is synchronized to the slot edge at chip rate. This means that a new symbol starts at the start of the slot (slot-edge=1).

Variable Amplifier 121

The complex symbol coming from the despreader is sent through the variable amplifier (VAMP) 121. The complete CCMR has one global AGC which sets the Pgain at slot rate. For different spreading factors, the initial gain must be set to a different value, for example, to 1.0 for SF 256, and to 64.0 for SF 4.

pilot filter 123. Slot wise coherent pilot symbol accumulation 124

In the pilot filter 123, a coherent pilot symbol accumulation 124 is done on a slot-by-slot basis. The Pac input defines if the symbol coming from the VAMP is a pilot symbol, as shown in Figure 16. In this example, the Psf is 256, and Pac would be 111100000 ....0000.

Pi with i=0,1,2, ..., the pilot symbol index, are the complex despread pilot symbols Dva (@fsymbB). In order to accumulate them coherently, the pilot modulation must be

removed first. This modulation is known a priori and must be present at the Psy input. For QPSK Psy can take four values: +i, -j, +j, -1.

For QPN Psy can take two values: +1 and -1. So, Psy is represented by a 2-bit value (Psy[0] and Psy[1]).

The values Pi are then demodulated in the following way (Piu are the demodulated values of Pi) (u= unmodulated):

Psy [0..1]	Pi	Piu
00	Pii + j * Piq	Pii + j * Piq
01	Pii + j * Piq	-Piq + j * Pii
10	Pii – j * Piq	Piq - j * Pii
11	Pii + j * Piq	-Pii - j * Piq

For QPN Psy must only take the values 00 or 11. Spj are the complex accumulations of these demodulated pilot symbols from the current slot, divided by the number of pilots (or multiplied by 1/number of pilot symbols): Sp = accumulation of Piu, divided by the number of pilot symbols.

This is equivalent to despreading over all the pilot chips in the slot in the case of unmodulated pilot symbols. Sp values are generated at slot rate fslot. The value is available at the end of the slot. This module is slot-synchronous.

Finger energy calculation 125

Here, a measure for the finger energy is calculated slot by slot. Because there is a delay of two slots on the chips, the energy is calculated from a delayed Sp value. This is shown in Figure 17. The energy is calculated as follows: $Sp\_i^2+Sq\_q^2$. With a delay of one slot on Sp. This energy will be used for the DLL and zero forcing.

Channel estimator 127

The channel estimator 127 performs a filtering or interpolation on the Sp values. The exact function to perform depends on the Chm (channel mode) input (fast or slow fading channels). The output of the channel estimator 127 is the channel estimation ces at the chip rate. When Chm = 0, the Ce_FIRcoef[4] and Ce_FIRmult[4] inputs are needed. When Chm = 1, the pipo input is needed.

Channel mode 0: Slow fading 131

In this mode, ces is constant over a complete slot. ces is a filtered version of the incoming Sp values as indicated in Figure 18. The multiplication after the filter is to have a FIR filter 129 with unity gain. To avoid a transient in the amplitude on the signal coming from the filter, four different values are stored for this gain. The first output of the filter gets gain CeFIRmult[0], the second output CeFIRmult[1], the third CeFIRmult[2] and CeFIRmult[3] is used on sample number 4 leaving the filter, and in steady state mode.

All filter taps should be initialized to 0 at the start of the process. The filter and multiplier work at slot rate fslot, and ces are samples at chip rate. (Oversampling of filter output). Figure 19 is an overview of the Rake finger process 131 in the case of channel mode 0.

The different pilot symbols are demodulated and coherently accumulated giving the values Sp0 to Sp5. The channel estimations ces are the output of the four taps FIR filter, ces0 is a function of Sp0 to Spa, and ces0 is constant over slot number 4. The De chip from slot 2 is delayed by two slots so that it is available with slot 4 as Dl chip. This chip is multiplied with the complex conjugate of ces0 to give the Dro chip of this finger.

The chip arriving in slot 2 is "corrected" with the information from the pilot symbols of the slot 0,1,2 and 3. Every chip is always corrected with the aid of the Before Before, Before, Present and After slot (unless some filter taps are set to 0). Channel estimations change only at slot rate. Note that Sp3 is generated together with the last chip of slot 3 while ces0, which is a function of Spa, is used for all chips of slot 4.

Channel mode 1: Fast fading 133 and 135

In this mode, ces are interpolated values between the current and the previous Sp values entering the channel estimator. Thus, ces changes at chip rate, as shown in Figure 20.

The incoming Sp values are positioned in the middle of the pilot portion to calculate the other complex values. The pipo (pilot position) input is used for this. It is an integer in the range [0:2559]. In Figure 21, pipo would be 768 or 769 (3/5*2560/2).

Linear interpolation is performed on both real and imaginary part of the Sp values. In this way we go via a straight line in the complex plane from Sp(k-1) to Sp(k). That is:

- Re[ces(i)]=(Re[Sp(k)]-Re[Sp(k-1)])*(i-pi_po)/2560+Re[Sp(k-1)]
- Im[ces(i)]=(Im[Sp(k)]-Im[Sp(k-1)])*(i-pi_po)/2560+Im[Sp(k-1)],

with i=0, 1, 2, ..., 2559. The 2560 different chips in a slot. See Figure 21 for an overview of the Rake finger process 135 in case of channel mode 1.

The different pilot symbols are demodulated and coherently accumulated giving the values Sp0 to Sp5. The channel estimations ces(i) for the chips i of slot 2 are calculated during slot 4 with the aid of Sp2 and Sp3. So, the Present and Future slot is used to make the channel estimates.

Channel correction 128 (Figure 15)

The channel correction 128 has as an input the delayed chips Dl coming from the FIFO and the channel estimations per chip ces. The function of the channel correction 128 is to correct for the channel phase of the finger and give a weight to the finger. The outputs from the different fingers can then be combined (coherently) in one signal. The following action is performed in these blocks: Dro = D1*ces(*), with ces(*) being the complex conjugates of ces.

Zero forcing 126 (Figure 15)

Each finger output can be forced to zero with the zf signal. The purpose of this is to set a finger to 0 when no (or very little) signal is present in that finger to avoid the accumulation of a lot of noise. The zf signal is obtained by comparing slot wise the FN and a programmable threshold, wherein zf is 1 if FN <= threshold.